

In the Claims:

Please amend claims 1, 2, 5 and 6 as follows:

1. (Currently Amended) A compiler device to generate an object code corresponding to a plurality of procedures in a source program, said plurality of procedures using an interface area in common, said object code for performing a procedure call in a program having a plurality of threads including a master thread and a non-master thread, making use of an interface area that is said plurality of procedures in parallel with a plurality of threads, one of said plurality of threads being a master thread and the others being non-master threads, said object code dynamically allocated allocating private interface areas corresponding to said interface area for said non-master threads, when said plurality of threads are processed in parallel.

2. (Currently Amended) A compiler device comprising:
a plurality of threads having a master thread and a non-master thread;
code generating means for generating a an object code for corresponding
to a plurality of procedures in a source program, said plurality of procedures using an
interface area in common, said object code being performed in parallel with a plurality of
threads, one of said plurality of threads being a master thread and the others being non-
master threads, said object code determining a leading address addresses of an private

~~interface area~~areas ~~that is~~which being dynamically allocated for said non-master ~~thread~~threads instead of the interface area used in common, when at least one of said plurality of threads is processed; and

code converting means for converting ~~directa-reference~~references to data ~~in the interface area in a source program~~the object code into a code for referencing the ~~interface area making use of~~indirect references to data in the private interface areas based on the leading ~~address~~addresses ~~determined by executing said code generated by said code~~generating means of the private interface areas.

3. (Previously Presented) A compiler device of claim 2,

wherein said code generating means is constituted to generate a code for calling a library for determining the leading address of the interface area that is dynamically allocated for each thread.

4. (Previously Presented) A compiler device of claim 2,

wherein said code generating means is constituted to generate a code for determining the leading address of the interface area designated by a user.

5. (Currently Amended) A computer-readable recording medium recorded

with a compiler program for causing a computer to realize a function to generate an object

~~code for performing a procedure call in a program having~~corresponding to a plurality of procedures in a source program, said plurality of procedures using an interface area in common, said object code being performed in parallel with a plurality of threads including, one of said plurality of threads being a master thread and a-the others being non-master threadthreads, making use of an interface area that issaid object code dynamically allocated allocating private interface areas corresponding to said interface area for said non-master threadthreads, when said plurality of threads are processed in parallel-are executed.

6. (Currently Amended) A computer-readable recording medium recorded with a compiler program ~~having a plurality of threads including a master thread and a non-master thread,~~ for causing a computer to realize:

~~a-an object code generating function for generating aan object code for~~corresponding to a plurality of procedures in a source program, said plurality of procedures using an interface area in common, said object code being performed with a plurality of threads, one of said plurality of threads being a master thread and others being non-master threads, said object code determining a-leading addressaddresses of an-private interface area that isareas which are dynamically allocated for the non-master threadthreads instead of the interface area used in common, when one of the plurality of threads is processed; and

a code converting function for converting ~~a referenced~~direct references to data in the interface area in a source program~~the object code~~ into a code for referencing the ~~interface area making use of~~indirect references to data in the private interface areas based on the leading address ~~determined by executing said code generated by said code generating function~~addresses of the private interfaces areas.

7. (Previously Presented) A computer-readable recording medium recorded with a compiler program of claim 6,

wherein said code generating function is constituted to generate a code for calling a library for determining the leading address of the interface area that is dynamically allocated for each thread.

8. (Previously Presented) A computer-readable recording medium recorded with a compiler program of claim 6,

wherein said code generating function is constituted to generate a code for determining the leading address of the interface area designated by a user.